[MEASURE ENERGY CONSUMPTION]

[PHASE_4]

- To create energy consumption visualizations, you'll need to have energy consumption data. These visualizations can help you identify trends, patterns, and anomalies in your energy usage.

- I can guide you through creating some common types of visualizations and provide examples of how to do it using tools like Python with libraries such as Matplotlib or Seaborn.

- Here are a few popular visualizations you might consider:

1. Time Series Line Plot:

- This plot shows energy consumption over time. You can create a line plot with time on the x-axis and energy consumption on the y-axis.

2. Bar Chart:

- A bar chart can show the energy consumption for different categories or time periods, such as monthly or yearly consumption.

3. Histogram:

- Histograms can help you understand the distribution of energy consumption values.

4. Box Plot:

- Box plots are useful for visualizing the distribution of energy consumption and identifying outliers.

5. Heatmap:

- A heatmap can show energy consumption patterns over time, helping you identify high and low consumption periods.

6. Scatter Plot:

- If you have multiple variables, you can create scatter plots to see if there are correlations between energy consumption and other factors.

Here's a simple example using Python and Matplotlib to create a time series line plot:

Python Program :

```
import matplotlib.pyplot as plt
```

```
import pandas as pd

# Load your energy consumption data into a DataFrame

# Replace 'data.csv' with your actual data file.

data = pd.read_csv('data.csv')

# Assuming your data has a 'timestamp' and 'consumption'
column.

# Convert the 'timestamp' column to a datetime object if it's not
already.

data['timestamp'] = pd.to_datetime(data['timestamp'])

# Create the line plot

plt.figure(figsize=(10, 6))

plt.plot(data['timestamp'], data['consumption'])

plt.xlabel('Time')

plt.ylabel('Energy Consumption')
```

plt.title('Energy Consumption Over Time')

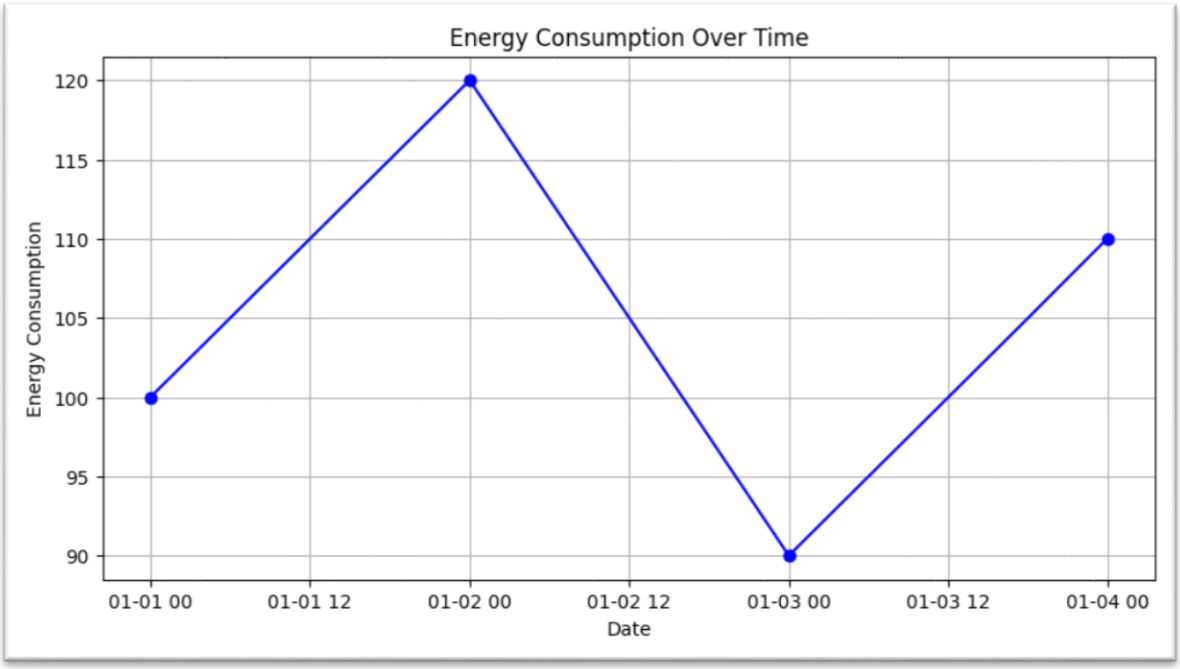
plt.grid(True)

plt.show()

EXAMPLE PROGRAM :

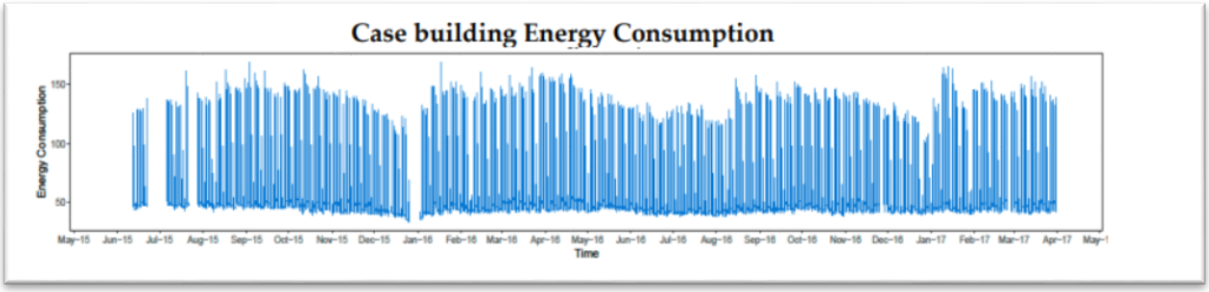
```
import matplotlib.pyplot as plt
import pandas as pd

# Sample energy consumption data (replace with your own dataset)
data = {
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-
04'],
    'Energy Consumption': [100, 120, 90, 110]
}

df = pd.DataFrame(data)
df['Date'] = pd.to_datetime(df['Date'])

plt.figure(figsize=(10, 5))
plt.plot(df['Date'], df['Energy Consumption'], marker='o',
linestyle='-', color='b')
plt.title('Energy Consumption Over Time')
plt.xlabel('Date')
plt.ylabel('Energy Consumption')
plt.grid(True)
plt.show()
```

OUTPUT :



Case Building and Case Dataset :

- The data comprises the electricity consumption and cooling thermal energy of a mixed function institutional building in a university campus in Singapore. Each data is measured at a half-hourly frequency from June 2015 to March 2017. As seen in the plots of the data (Figure 1), there is missing data in both datasets.



The energy consumption data exhibits weekly cycles (Figure 2). The plot shows the energy consumption for a typical week. Energy consumption peaks during office hours and reduces to a minimum in the hours of the night and early morning. The energy consumption is also lower on Saturdays and lowest on Sundays. The plot also shows a clear correlation between the energy consumption and cooling capacity in use.

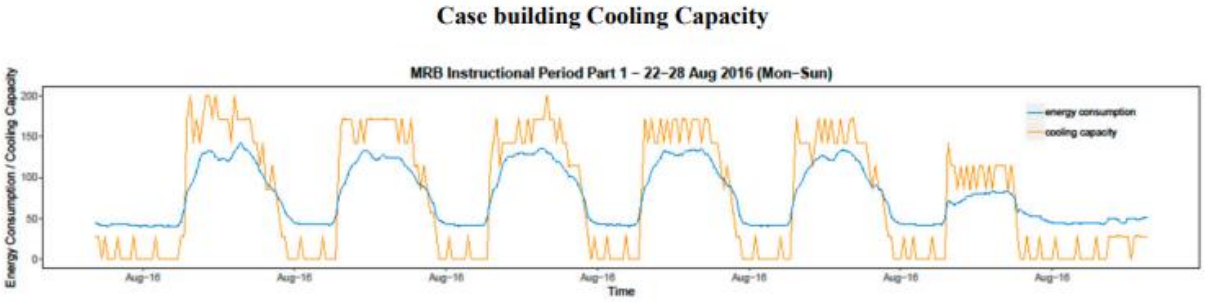


Figure 2. Weekly cycles in the data.

Data Transformation :

- Data transformation in energy consumption refers to the process of modifying or converting raw energy consumption data into a more useful and informative format. This transformation is often necessary to make the data suitable for analysis, visualization, or modeling.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Sample energy consumption data in a CSV file

data_file = "energy_consumption_data.csv"

# Load the data into a Pandas DataFrame

df = pd.read_csv(data_file)
```

```
# View the first few rows of the data

print("Original Data:")

print(df.head())

# Data Transformation Steps

# 1. Aggregation - Convert hourly data to daily data by summing consumption

df['Date'] = pd.to_datetime(df['Timestamp'])

daily_data = df.resample('D', on='Date').sum()

# 2. Normalization - Scale the data using min-max scaling

daily_data['Consumption (Normalized)'] = (daily_data['Consumption'] - daily_data['Consumption'].min()) / (daily_data['Consumption'].max() - daily_data['Consumption'].min())

# 3. Data Visualization - Plot the original and normalized data

plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)

plt.plot(daily_data['Date'], daily_data['Consumption'])

plt.title("Original Daily Consumption")

plt.subplot(2, 1, 2)

plt.plot(daily_data['Date'], daily_data['Consumption (Normalized)'])

plt.title("Normalized Daily Consumption")

plt.tight_layout()

plt.show()
```

4. Seasonal Decomposition - Decompose the data into seasonal, trend, and residual components

```
from statsmodels.tsa.seasonal import seasonal_decompose

result = seasonal_decompose(daily_data['Consumption'], model='additive', freq=365)  # Assumes a yearly seasonality
```

5. Data Visualization - Plot the decomposition components

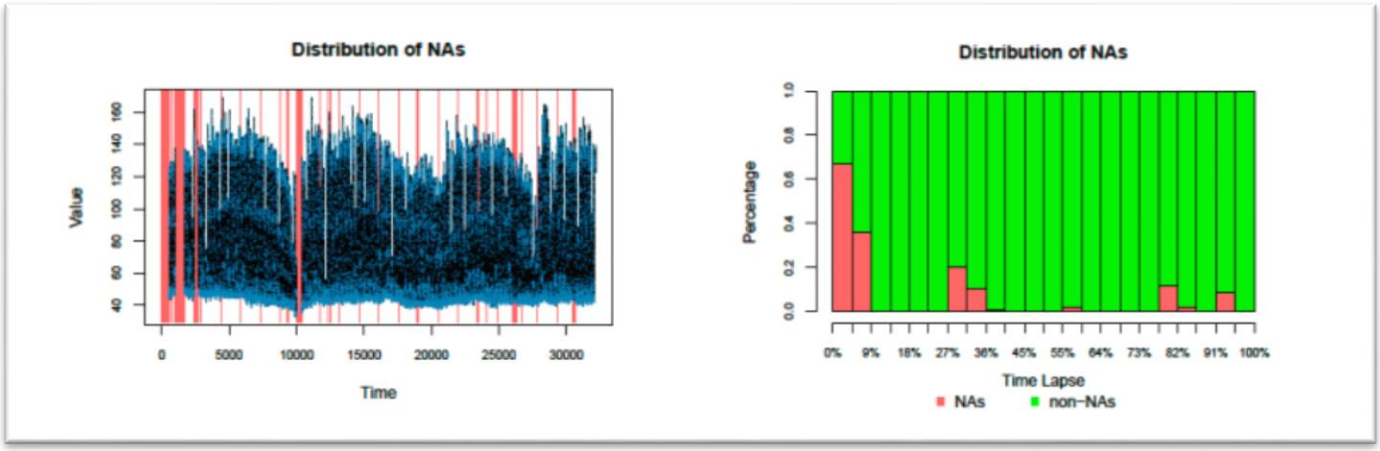
```
plt.figure(figsize=(12, 6))

result.plot()

plt.show()
```

6. Data Categorization - Categorize data into different sectors

```
df['Sector'] = pd.cut(df['Consumption'], bins=[0, 500, 1000, 1500], labels=['Low', 'Medium', 'High'])
```

View the first few rows of the categorized data

```
print("Categorized Data:")

print(df.head())
```



	bankname	bank	year	quarter	quarters	beta	leverage	roa	r_rwa	rwa_assets
178	Sparebank SMN	3	2011	4	2011q4	.7119	12.9143	.00277	.003803	.739655
179	Sparebank SMN	3	2012	1	2012q1	.0361	12.528	.002714	.003588	.773668
180	Sparebank SMN	3	2012	2	2012q2	.6157	12.3613	.002302	.003043	.740516
181	Sparebank SMN	3	2012	3	2012q3	.3987	12.5357	.002801	.003756	.751244
182	Sparebank SMN	3	2012	4	2012q4	.4382	11.5395	.002388	.003153	.763566
183	Sparebank SMN	3	2013	1	2013q1	.804	11.436	.002935	.00389	.745497
184	Sparebank Vest	4	1998	1	1998q1	.4144	.	.	.	.
185	Sparebank Vest	4	1998	2	1998q2	.1306	.	.	.	.
186	Sparebank Vest	4	1998	3	1998q3	.1818	.	.	.	.
187	Sparebank Vest	4	1998	4	1998q4	.3931	.	.	.	.
188	Sparebank Vest	4	1999	1	1999q1	-.3533	.	.004946	.	.
189	Sparebank Vest	4	1999	2	1999q2	.4742	15.9602	.002861	.004298	.66559
190	Sparebank Vest	4	1999	3	1999q3	-.113	.	.002546	.	.
191	Sparebank Vest	4	1999	4	1999q4	.4135	14.2458	.004057	.006104	.648981
192	Sparebank Vest	4	2000	1	2000q1	.1378	.	.002616	.	.
193	Sparebank Vest	4	2000	2	2000q2	.0917	15.2056	.00157	.002422	.64823
194	Sparebank Vest	4	2000	3	2000q3	-.1545	15.6238	.00257	.003951	.652837
195	Sparebank Vest	4	2000	4	2000q4	.2499	15.2741	.001703	.002639	.638658
196	Sparebank Vest	4	2001	1	2001q1	.4581	15.2077	.000838	.001301	.649364
197	Sparebank Vest	4	2001	2	2001q2	.2473	15.5972	.001753	.002712	.64303
198	Sparebank Vest	4	2001	3	2001q3	.674	15.7925	.001156	.001774	.660849
199	Sparebank Vest	4	2001	4	2001q4	.0563	15.146	.000589	.000876	.682053
200	Sparebank Vest	4	2002	1	2002q1	-.0516	15.4266	.002142	.003115	.693378

:

- SVR is used as the benchmarking base model for comparison of the gated neural network, where
- SVR is unable to handle missing values in training data. Discarding data has two disadvantages.
- A resulting smaller training dataset may adversely affect the performance of the trained model.

In addition, if the missing data mechanism is not MCAR , its removal will introduce biases into the training dataset.

Given the results in Table 2 root mean square error (RMSE) for data imputation (lowest RMSE highlighted), the structural model method was selected for missing data imputation.

Missing data of larger gap size were not imputed.

Neural network models handle missing input data in another way. In this phase of modeling,

the remaining missing data were replaced by a value not observed in the data. The selected value

is –1 as the input data was normalized to [0, 1] using min–max scaling during pre-processing. Only data samples in which the label or all the inputs are missing, were discarded .

	bankname	bank	year	quarter	quarters	beta	leverage	roa	r_rwa	rwa_assets
178	Sparebank SMN	3	2011	4	2011q4	.7119	12.9143	.00277	.003803	.739655
179	Sparebank SMN	3	2012	1	2012q1	.0361	12.528	.002714	.003588	.773668
180	Sparebank SMN	3	2012	2	2012q2	.6157	12.3613	.002302	.003043	.740516
181	Sparebank SMN	3	2012	3	2012q3	.3987	12.5357	.002801	.003756	.751244
182	Sparebank SMN	3	2012	4	2012q4	.4382	11.5395	.002388	.003153	.763566
183	Sparebank SMN	3	2013	1	2013q1	.804	11.436	.002935	.00389	.745497
184	Sparebank Vest	4	1998	1	1998q1	.4144	.	.	.	.
185	Sparebank Vest	4	1998	2	1998q2	.1306	.	.	.	.
186	Sparebank Vest	4	1998	3	1998q3	.1818	.	.	.	.
187	Sparebank Vest	4	1998	4	1998q4	.3931	.	.	.	.
188	Sparebank Vest	4	1999	1	1999q1	-.3533	.	.004946	.	.
189	Sparebank Vest	4	1999	2	1999q2	.4742	15.9602	.002861	.004298	.66559
190	Sparebank Vest	4	1999	3	1999q3	-.113	.	.002546	.	.
191	Sparebank Vest	4	1999	4	1999q4	.4135	14.2458	.004057	.006104	.648981
192	Sparebank Vest	4	2000	1	2000q1	.1378	.	.002616	.	.
193	Sparebank Vest	4	2000	2	2000q2	.0917	15.2056	.00157	.002422	.64823
194	Sparebank Vest	4	2000	3	2000q3	-.1545	15.6238	.00257	.003951	.652837
195	Sparebank Vest	4	2000	4	2000q4	.2499	15.2741	.001703	.002639	.638658
196	Sparebank Vest	4	2001	1	2001q1	.4581	15.2077	.000838	.001301	.649364
197	Sparebank Vest	4	2001	2	2001q2	.2473	15.5972	.001753	.002712	.64303
198	Sparebank Vest	4	2001	3	2001q3	.674	15.7925	.001156	.001774	.660849
199	Sparebank Vest	4	2001	4	2001q4	.0563	15.146	.000589	.000876	.682053
200	Sparebank Vest	4	2002	1	2002q1	-.0516	15.4266	.002142	.003115	.693378

Forecasting Methodology:

The typical machine learning workflow comprises the following steps:

1. Define problem and measure of success;

2. Define an evaluation protocol;

3. Prepare data;

4. Develop benchmark and base models;

5. Scale up and regularize base model.

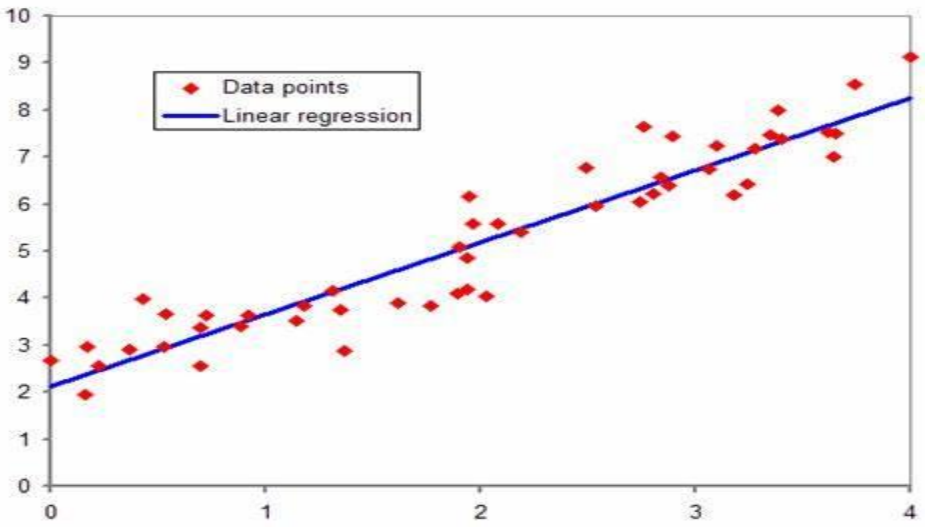
- This basic workflow applies to both shallow models as well as deep. Steps 1 and 2 are largely self-explanatory. In this study, the problem is energy load forecasting and a suitable metric is specified in a later section, Forecasting Task. The standard evaluation protocol used in machine learning is cross-validation.
- In this study, due to limitations in the availability of computation resources, an out-of-sample or last-block evaluation approach is used instead. The time series dataset is partitioned using a simple 2:1 split with the larger split forming the training split and the other the validation split.

- The main difference in workflow between shallow and deep models lies in the data preparation step. In this step, carefully crafted manual features which help shallow models perform well are introduced into the modeling

process. Deep learning model obviate the need for such intensive and manual feature engineering work by learning the requisite higher-level representations for successfully performing the machine learning task automatically and independently

Forecasting Task:

- The forecasting task for this analysis is a one-step or 30-minute ahead energy consumption forecast
- Using a 5-day lookback which is a sequence of 240-time steps (i.e., 48-time steps per day × 5 days).
- The sequence length is selected given the weekly cycle seen in the data.



- At least 5 days will allow the current day of the week to be determined given the weekly cycle

- H-H-H-H-H-L-LL where H is high consumption, L is low, and LL lowest. For example, for a sequence
- H-H-H-H-H, the next value in the sequence is L.
- The metric used in the comparison analysis is mean absolute error (MAE).

Conclusion :

- While this work has raised questions regarding building energy data forecasting, it is computationally intensive to test on each model, especially when more data will be used to train the model.
- One future work is to look at the application of transfer learning in the building energy data forecasting to reduce the computational time when applying the same method to more buildings or meters.
- Another direction is to de-trend, or de-seasonalize data according to building function and occupancy patterns, which could increase the accuracy and reduce additional computational time.